

09/550,192
Art Unit 3624
8446.00

following sentence:

-- This Application has been assigned serial
number 09/550,191 and a filing date of April
17, 2000. --

REMARKS

This Amendment is a Corrected Amendment, submitted in response to the Office Action mailed on August 27, 2003. That Office Action responded to the original Amendment submitted in response to the Office Action mailed on January 30, 2003.

This Corrected Amendment corrects supposed defects in the original Amendment. Claims 1 - 7 are pending, and all stand rejected at present.

Claims 8 - 13 were added. Support is found in the Specification, at least beginning on page 31, section entitled "One Form of Invention." For example, the last sentence of the first paragraph in this section refers to modifying software in a payment switch, and installing the modified software.

The Specification has been amended to further identify a patent application which was incorporated by reference.

Response to 112 - Rejections

Claim 1

Claim 1 has been amended.

09/550,192
Art Unit 3624
8446.00

In response to the rejection of claim 1(b)(iii), based on the term "at least one," Applicant requests a citation of authority in support of the rejection.

One reason is that the reason given by the Office Action is that the claim language is "open ended." However, being "open ended" does not, by itself, render a claim indefinite. Indefiniteness is the basis of any 112 - rejection. That is, "at least one" covers all numbers from one to infinity. There is nothing indefinite about that. It may be a large collection of numbers, but it is, in fact, definite.

If the rejection is, in effect, asking "how many modules EXACTLY are being claimed ?" Applicant points out that an exact number need not be specified. The answer is "at least one."

Further, it is axiomatic that the claims are read as-a-whole. The phrase "at least one" is read with the rest of the claim, not in isolation.

Further still, the undersigned attorney did a search of the PTO's database, looking for the phrase "at least one" in patents issued from 1976 to the present. Almost one million hits were returned: 919,632 hits, to be precise.

Applicant fails to see how this rejection can be justified, in view of the fact that, since 1976, almost one million patents have used the claim language in question.

09/550,192
Art Unit 3624
8446.00

Claims 6 and 7

Applicant respectfully submits that the PTO mis-applies the law of claim interpretation.

The question which the claim must answer is **NOT** this: "Which specific ones of the PAK_MOD modules contain unit B, with no unit C ?"

Rather, the question is "Do **SOME** of the PAK_MOD modules contain unit B, with no unit C ?" If so, then the claim recitation (6(b)(iii)(B) in this case) is found.

From another point of view, Applicant **does not know** how an infringer will choose to arrange the modules. For that reason alone, Applicant cannot say "which" modules will be covered by the claim language.

RESPONSE TO 102 - REJECTIONS

Claim 1

Claim 1 recites:

1. A method of constructing a plurality of software systems, comprising the following steps:

a) maintaining an inventory of software modules, which includes:

i) a group of type A modules; and

ii) a collection of type B modules;

b) when constructing each software system,

09/550,192
Art Unit 3624
8446.00

- i) including copies of the entire group of type A modules;
- ii) including copies of [selected] some or all type B modules; and
- iii) generating at least one customized module, which is a copy of neither a type A nor a Type B module.

Claim 1(b)(i) and (ii)

The Office Action relies on Yates, column 18, lines 1 - 13 to show claim 1(b)(i) and (b)(ii). Those claim passages state that

- 1) all of the type A modules are included in the software system being constructed and
- 2) some or all of the type B modules are included.

That passage of Yates relied on by the PTO is here set forth:

Known constructions, where policies¹ are embedded in the objects, require rewriting of code in the object to change behavior.

External policies allow not only changes in behavior to be achieved more easily but also more freely, and can allow extra behaviors (which are composed from combinations/permutations of a programmed set of operations) to be performed even if these were not originally anticipated.

¹ "Policy" is apparently jargon for a section of computer code.

09/550,192
Art Unit 3624
8446.00

The concept of policies is such that an object must have access to a "Policy Interpreter."

This can be internal or external to the object.

In order to locate policies, a policy server might be provided, again either internal or external to an object.

(Yates, column 18, lines 1 - 13.)

Plainly, the Yates passage has no relevance to claim 1(b)(i) and (b)(ii). This passage fails to show anything corresponding to type A and type B modules, nor the particular selections recited in claim 1(b)(i) and (ii).

MPEP § 2131 states:

A claim is anticipated only if **each and every element** as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.

Applicant submits that claim 1(b)(i) and (b)(ii) have not been shown in Yates, and thus claim 1 cannot be anticipated.

In addition, Applicant points out that it is axiomatic that, in order to **anticipate** claim 1, the Yates reference must **infringe** claim 1. (See PATENTS, A Treatise on the Law of Patentability, Validity, and Infringement, by D. Chisum, section 3.02[1], entitled, "The Classic Infringement Test.")

Applicant submits that the Yates passage set out above clearly

09/550,192
Art Unit 3624
8446.00

does not show claim 1(b)(i) and (b)(ii).

Therefore, Applicant submits that Yates does not anticipate claim 1.

Claim 1(a)

Claim 1(a) recites maintaining type A and type B modules. The Office Action relies on identical passages in Yates to show both of these module types. Those passages are

- 1) The Abstract
- 2) Column 2, lines 57 - 65,
- 3) Column 4, lines 3 - 12,
- 4) Column 5, lines 40 - 55, and
- 5) Column 18, lines 1 - 13.

As to item (1), Yates' Abstract merely refers to selecting "reusable software modules." There is no reference to anything analogous to types A and B.

As to item (2), the passage merely refers to selecting a "set" of software modules.

Item (3) merely refers to "adding" or "modifying" software modules.

Item (4) refers to changing a set of software modules which is used at a given time.

Item (5) refers to the Yates passage which was quoted above. That passage refers to modifying a "set" of software modules.

09/550,192
Art Unit 3624
8446.00

Therefore, Applicant submits that the two "types" of module recited in claim 1(a) have not been shown in Yates.

Claim 1(b)(iii)

To show claim 1(b)(iii), the Office Action relies on the five identical passages used to reject claims 1(b)(i) and (ii). Those passages, summarized immediately above, fail to show

iii) generating at least one customized module, which is a copy of neither a type A nor a Type B module,

as recited in claim 1(b)(iii).

Claims 2 - 5

Claims 2 - 5 depend from claim 1, and are considered allowable based on claim 1.

For purposes of 37 CFR § 111, this statement is deemed sufficient, since, by law, if claim 1 is patentable, then all its dependant claims are also patentable.

Claim 2

In addition, claim 2 recites:

2. Method according to claim 1, wherein each system constructed performs the following functions:

09/550,192
Art Unit 3624
8446.00

- 1) processing of the content of messages;
- 2) packaging of messages into packets for transport out of the system;
- 3) transfer of messages into, and out of, the system; and
- 4) coordination of functions (1), (2), and (3).

CLAIM 2(1)

The Office Action relies on four passages in Yates to show claim 2(1), namely:

- column 1, lines 1 - 5;
- column 2, lines 57 - 65;
- column 4, lines 3 - 12; and
- column 16, lines 11 - 23.

However, claim 2(1), together with its context, states that

. . . each system constructed performs the following functions:

- 1) processing of the content of messages.

The first passage cited in Yates says nothing about whether "each system constructed" performs a given function.

The second passage states that "A reconfigurable software agent **MAY** comprise . . ." That is inconsistent with claim 2(1). If claim 1(1) were shown in that passage, then the passage would

09/550,192
Art Unit 3624
8446.00

state that the reconfigurable software agent "**MUST** comprise," or equivalent.

The third passage merely lists software modules which **may** be available. Again, that does not show claim 2(1).

The fourth passage merely lists some functions which a "service retailer domain 103" performs. That does not show claim 2(1).

Therefore, the mandatory language of claim 2 ("**each** system constructed performs the following functions . . .") has not been shown in Yates.

CLAIM 2(2)

The Office Action relies on two passages in Yates to show claim 2(2), namely:

column 14, lines 37 - 51; and

column 18, lines 9 - 13.

Claim 2(2) recites:

. . . wherein **each system constructed** performs the following functions:

. . .

2) packaging of messages into packets for transport out of the system;

The first cited passage in Yates merely refers to a type of communication. But it does not state that "each system constructed

09/550,192
Art Unit 3624
8446.00

. . ." possesses that communication. Thus, even if the type of communication shown in Yates corresponds to claim 2(2), claim 2(2) cannot be read out of context.

The second passage states:

The concept of policies is such that an object must have access to a "Policy Interpreter." This can be internal or external to the object. In order to locate policies, a policy server might be provided, again either internal or external to an object.

(Column 18, lines 9 - 13.)

The second passage clearly does not relate to claim 2(2).

Therefore, the mandatory language of claim 2 ("each system constructed performs the following functions . . .") has not been shown in Yates.

CLAIM 2(3)

The Office Action cites one passage in Yates to show claim 2(3), namely, column 5, lines 40 - 55. Claim 2(3) recites:

. . . wherein each system constructed performs the following functions:

. . .

3) transfer of messages into, and out of, the system.

The cited passage in Yates says absolutely nothing about

09/550,192
Art Unit 3624
8446.00

transferring messages into and out of the system, together with requiring that "each system constructed" does that transfer.

Therefore, the mandatory language of claim 2 ("**each** system constructed performs the following functions . . .") has not been shown in Yates.

CLAIM 2(4)

The Office Action cites one passage in Yates to show claim 2(4), namely, column 4, lines 3 - 12.

Claim 2(4) recites:

. . . wherein each system constructed performs
the following functions:

. . .

4) coordination of functions (1),
(2), and (3).

As explained above, claim 2(1), (2), and (3) are not found in Yates. Thus, the passage in question, namely, column 4, lines 3 - 12, cannot refer to "coordination" of the functions of those parts of the claim.

In addition, the passage in question (column 4, lines 3 - 12) does not refer to the other passages cited in Yates to show the other parts of claim 2. Thus, it would seem conclusive that the passage in question (column 4, lines 3 - 12) does not show the claimed coordination, since it does not refer to coordinating

09/550,192
Art Unit 3624
8446.00

elements in the other passages cited.

Therefore, Applicant submits that claim 2 is not shown in Yates.

Claim 3

Claim 3 recites:

3. Method according to claim 2, wherein functions (3) and (4) are performed using **type A modules** exclusively.

"Type A modules" are those recited in claim 1. Claim 1 states that **all** type A modules in the group are included in every system constructed.

Claim 3 depends from claim 2, which defines "functions (2) and (3)." The passages cited in Yates to show those functions (2) and (3) of claim 2 do not refer to "type A modules."

In addition, the passage used to reject claim 3 is column 18, lines 1 - 13. That passage of Yates relied on by the PTO is here set forth:

Known constructions, where policies² are embedded in the objects, require rewriting of code in the object to change behavior.

External policies allow not only changes in behavior to be achieved more easily but also

² "Policy" is apparently jargon for a section of computer code.

09/550,192
Art Unit 3624
8446.00

more freely, and can allow extra behaviors (which are composed from combinations/permutations of a programmed set of operations) to be performed even if these were not originally anticipated.

The concept of policies is such that an object must have access to a "Policy Interpreter."

This can be internal or external to the object.

In order to locate policies, a policy server might be provided, again either internal or external to an object.

(Yates, column 18, lines 1 - 13.)

Plainly, this passage does not show claim 3.

Claim 4

Claim 4 recites:

4. Method according to claim 3, wherein function (1) is performed using a combination of type A, type B, and customized modules.

The Office Action relies on two passages in Yates to show claim 4, namely,

column 4, lines 26 - 35 and

column 18, lines 1 - 13.

The latter passage was cited in full in the previous section, concerning claim 3, and plainly does not show claim 4, as a side-by-side comparison indicates.

The former passage states that "at least one of the software

09/550,192
Art Unit 3624
8446.00

agents" is provided with certain functionality. If the "software agent" is considered to correspond to the "system" of the claims, and the undersigned attorney sees no other element in the passage which can correspond to the "system," then claim 4 is clearly not shown. Claim 4, through its parent claim(s), states that "each system constructed" is equipped with the enumerated functions. The passage in question merely states that "at least one software agent" is equipped with certain functions.

That does not show claim 4.

Claim 5

Claim 5 recites:

5. Method according to claim 4, wherein function (2) is performed using a combination of type A, type B, and customized modules.

The identical passages in Yates cited to show claim 4 were cited to show claim 5.

One passage is cited in full, in the section above regarding claim 3. A side-by-side comparison clearly indicates that the passage does not show claim 5.

The other passage (column 4, lines 26 - 35) states that "at least one of the software agents" is provided with certain functionality. If the "software agent" is considered to correspond to the "system" of the claims, and the undersigned attorney sees

09/550,192
Art Unit 3624
8446.00

no other element in the passage which can correspond to the "system," then claim 5 is clearly not shown.

Claim 5, through its parent claim(s), states that "each system constructed" is equipped with the enumerated functions. The passage in question merely states that "at least one software agent" is equipped with certain functions.

That does not show claim 5.

Claim 7

Claim 7 recites:

7. Method according to claim 6, and further comprising the following step:

iv) fabricating PROC_MOD modules in all systems, such that:

- A) copies of a software unit D is contained in every PROC_MOD module;
- B) some PROC_MOD modules contain a software unit E with no unit F; and
- C) some PROC_MOD modules contain a software unit F with no unit E.

The same two passages in Yates, cited to show claims 4 and 5 were cited to show (A), (B), and (C) of claim 7(iv). Those passages are

column 4, lines 26 - 35 and
column 18, lines 1 - 13.

09/550,192
Art Unit 3624
8446.00

The latter passage is here repeated:

Known constructions, where policies³ are embedded in the objects, require rewriting of code in the object to change behavior.

External policies allow not only changes in behavior to be achieved more easily but also more freely, and can allow extra behaviors (which are composed from combinations/permutations of a programmed set of operations) to be performed even if these were not originally anticipated.

The concept of policies is such that an object must have access to a "Policy Interpreter."

This can be internal or external to the object.

In order to locate policies, a policy server might be provided, again either internal or external to an object.

(Yates, column 18, lines 1 - 13.)

A side-by-side comparison between each recitation in claim 7 and the cited passage clearly shows that the cited passage does not show the claim elements. For example, claim 7(iv)(C) states:

C) some PROC_MOD modules contain a software unit F with no unit E.

The absence of the unit E is not discussed in this passage. That, by itself, is sufficient to preclude the rejection.

³ "Policy" is apparently jargon for a section of computer code.

09/550,192
Art Unit 3624
8446.00

The other passage (column 4, lines 26 - 35) states that "at least one of the software agents" is provided with certain functionality. If the "software agent" is considered to correspond to the "system" of the claims, and the undersigned attorney sees no other element in the passage which can correspond to the "system," then claim 7 is clearly not shown.

Claim 7, through its parent claim(s), states that "each system constructed" is equipped with the enumerated functions. The passage in question merely states that "at least one software agent" is equipped with certain functions.

That does not show claim 7.

Re: Claims 2 - 5 and 7

MPEP § 2131 states:

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.

The passages cited in Yates do not appear to show the claim recitations for which they were cited. Therefore, Applicant requests that the specific claim recitations be identified in Yates.

37 CFR § 1.104(c)(2) states:

. . . the Examiner must cite the best

09/550,192
Art Unit 3624
8446.00

references at his . . . command.

When a reference is complex or shows or describes inventions other than that claimed by the applicant, the particular part relied on must be designated as nearly as practicable.

The pertinence of each reference, if not apparent, must be clearly explained and each rejected claim specified.

Since, as explained above, all cited passages of Yates contain material "other than that claimed by Applicant," this regulation requires a more precise identification of the anticipating material in Yates.

Claim 6

Claim 6(a)

Applicant points out that claim 6(a) recites:

a) fabricating a collection of software systems, each of which contains

and then lists four types of module.

Restated, claim 6(a) states that every "software system" in the "collection" contains (at least) the four modules listed in 6(a)(i) through (a)(iv).

The Office Action cites two passages in Yates to show this, namely,

1) column 2, lines 38 - 65

09/550,192
Art Unit 3624
8446.00

and

2) column 4, lines 13 - 65.

However, those passages contain nothing more than generalized statements indicating that, in different situations, systems may be designed which are different.

That is **directly contrary** to the claim recitations in question. One reason is that claim 6(a) states that every software system contains four specific modules. Thus, in that respect, every software system is **identical**.

That is contrary to the cited passages in Yates.

Claim 6(b)

Claim 6(b) states that all of the "software systems" of claim 6(a) will contain two elements, namely (1) identical CONTROL modules and (2) identical COM_MOD modules. The Office Action relies on two passages in Yates to show this.

One passage is column 4, lines 26 -35. However, that passage merely states that "at least one . . . software agent" is equipped with certain functionality. That does not show the claim recitations in question, and is actually inconsistent with it.

It is inconsistent because the Yates-passage only focuses on **ONE** software agent, and lists some properties of that agent. It fails to identify a **GROUP** of agents. Claim 6 refers to properties of a "collection of software systems." Yates' discussion of a

09/550,192
Art Unit 3624
8446.00

SINGLE software agent does not show the properties of a **GROUP** as in claim 6.

The other passage relied on by the PTO is Yates column 18, lines 1 - 13. That passage is set out verbatim above, and clearly does not show claim 6(b)(i) and (b)(ii).

Claim 6(b)(iii)

Applicant points to claim 6(b)(iii), which is repeated here:

iii) fabricating PAK_MOD modules in all systems, such that:

- A) copies of a software unit A is contained in every PAK_MOD module;
- B) some PAK_MOD modules contain a software unit B with no unit C; and
- C) some PAK_MOD modules contain a software unit C with no unit B.

The undersigned attorney has examined the passages in Yates which are cited to show these recitations, and cannot locate the recitations in those passages. [Actually, the passages used by the PTO are the same as used for claim 6(b)(i) and (b)(ii).]

Further Consideration of Claim 6

Claim 6(a)(iii) refers to a "communications module (COM_MOD) which accepts and delivers message packets." Yates, column 14,

09/550,192
Art Unit 3624
8446.00

line 49 et seq., refers to a Manager which uses data packets for file transfer. It is Assumed arguendo that Yates' Manager shows the recited COM_MOD.

Claim 6(b)(ii) states that, during fabrication of software systems, "fabricating identical COM_MOD modules in all systems." The Office Action relies on two passages in Yates to show this.

One passage is column 4, lines 26 - 35. However, that passage, in essence, states that each "software agent" is "customized" for a specific purpose. Thus, the agents will be different. That does not state, or even imply, "identical COM_MOD modules in all" agents.

The second passage is column 18, lines 1 - 13. That passage was set forth verbatim above. That passage merely refers to a process of modifying software. That does not state, or even imply, "identical COM_MOD modules in all" agents.

In fact, it would tend to show the opposite. If software (ie, the COM_MOD modules) is modified, then for "identical COM_MOD modules" to exist in all agents, all those modules must be modified in **the same manner**. That has not been shown in Yates.

General Observations on Yates

If an attempt is made to apply claim 1 to Yates, the undersigned attorney believes that the only possible elements in Yates which apply to (1) the "software system," (2) the type A

09/550,192
Art Unit 3624
8446.00

modules, and (3) the type B modules of claim 1 are the following, respectively:

- The "agents" (corresponding to the "software system"),
- The SIBBs, Service Independent Building Blocks (corresponding to the type A modules), and
- The "adaptors" (corresponding to the type B modules.)

(See column 17, lines 12 - 22.)

However, several problems arise. Claim 1(b)(i) states that "the entire group" of the type A modules is included in the "software system." Yates expressly states that is not so. He states that the SIBBs in the "agents" change over time. (Column 18, lines 42 - 48.) Thus, as a minimum, the "entire group" of the SIBBs does not remain constant.

An the undersigned attorney can find no statement that "the entire group" of the SIBBs is given to each "agent" in the first place.

Another problem arises in connection with claim 3, which states:

3. Method according to claim 2, wherein functions (3) and (4) are performed using type A modules exclusively.

09/550,192
Art Unit 3624
8446.00

"Functions (3) and (4)" of claim 2 refer to transfer of messages into, and out of, the system. Yates' SIBBs do not do that. His Communications Session Manager does that. (Column 14, line 49 et seq.) That Manager does not appear to be a SIBB.

Therefore, Applicant requests that the "group of type A modules" be identified in Yates.

Response to RESPONSE TO ARGUMENTS

As to the Yates passage, the passage merely refers to modifying computer code. The claim language in question does not recite that.

The language "at least" is accurate: the claim is a "comprising" type of claim. Other elements can be present in an infringing device. Further, it appears that the Office Action is demanding that Applicant submit an amendment to an amendment. That is not allowed.

As to In re Haza, even if the rule stated by the Office Action be correct, that rule only applies when the claim elements in question have been shown in the prior art. That has not been done here.

Further, the stated rule is that "a plurality of elements has no patentable significance unless a new and unexpected result is produced." That statement is necessarily incorrect. It is

09/550,192
Art Unit 3624
8446.00

contrary to section 102, which states that a "novel" combination of old elements is patentable (if non-obvious, of course).

Further still, the "rule" is inapplicable here. All claims are method claims.

Added Claims

Claim 8

Claim 8 depends from claim 1, and recites "c) installing the software systems into electronic payment switches." That has not been shown in the prior art.

Claim 9

Claim 9 depends from claim 2, and recites "c) installing the software systems into electronic payment switches." That has not been shown in the prior art.

Claim 10

Claim 10 depends from claim 6, and recites "c) installing the software systems into electronic payment switches." That has not been shown in the prior art.

Claim 11

Claim 11 depends from claim 7, and recites "c) installing the software systems into electronic payment switches." That has not

09/550,192
Art Unit 3624
8446.00

been shown in the prior art.

Claim 12

Claim 12 depends from claim 1, and recites "d) installing the software systems into electronic payment switches," plus another recitation. However, recitation (d) has not been shown in the prior art, and that, by itself, is sufficient to preclude rejection.

Claim 13

Claim 13 depends from claim 6, and recites "d) installing the software systems into electronic payment switches," plus another recitation. However, recitation (d) has not been shown in the prior art, and that, by itself, is sufficient to preclude rejection.

Re: All Added Claims

Applicant points out that the absence of the elements identified above from the prior art is sufficient to preclude rejection. If the rejection is 102-type, that absence is conclusive.

If the rejection is 103-type, that absence shifts the burden to the PTO, who must then show a prima facie case of obviousness. Until that is done, the claims are deemed patentable.

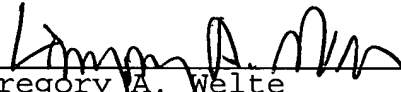
09/550,192
Art Unit 3624
8446.00

Conclusion

Applicant requests that the rejections to the claims be reconsidered and withdrawn.

Applicant expresses thanks to the Examiner for the careful consideration given to this case.

Respectfully submitted,


Gregory A. Welte
Reg. No. 30,434

NCR Corporation
1700 South Patterson Blvd.
WHQ - 5E
Dayton, OH 45479
September 29, 2003
(937) 445 - 4956

ATTACHMENTS:

1. Annotated Claim(s) Showing Amendments (per previous rules)
2. Current Status of all claims (per current rules)

09/550,192
Art Unit 3624
8446.00

ATTACHMENTS:

Annotated Claim(s) Showing Amendments (per previous rules)

1. A method of constructing a plurality of software systems, comprising the following steps:

a) maintaining an inventory of software modules, which includes:

- i) a group of type A modules; and
- ii) a collection of type B modules;

b) when constructing each software system,

i) including copies of the entire group of type A modules;

ii) including copies of [selected] some or all type B modules;

and

iii) generating at least one customized module, which is a copy of neither a type A nor a Type B module.

Current Status of all Claims (per current rules)

Comment: the parenthetical status of each claim refers to the status at the time of the Amendment which this Corrected Amendment replaces. One reason is that the last Office Action does not state that the last Amendment, which this Corrected Amendment replaces,

09/550,192
Art Unit 3624
8446.00

was entered.

1. (Currently amended) A method of constructing a plurality of software systems, comprising the following steps:

a) maintaining an inventory of software modules, which includes:

- i) a group of type A modules; and
- ii) a collection of type B modules;

b) when constructing each software system,

- i) including copies of the entire group of type A modules;
- ii) including copies of selected type B modules; and
- iii) generating at least one customized module, which is a copy of neither a type A nor a Type B module.

2. (Original) Method according to claim 1, wherein each system constructed performs the following functions:

- 1) processing of the content of messages;
 - 2) packaging of messages into packets for transport out of the system;
 - 3) transfer of messages into, and out of, the system;
- and

09/550,192
Art Unit 3624
8446.00

4) coordination of functions (1), (2), and (3).

3. (Original) Method according to claim 2, wherein functions (3) and (4) are performed using type A modules exclusively.

4. (Original) Method according to claim 3, wherein function (1) is performed using a combination of type A, type B, and customized modules.

5. (Original) Method according to claim 4, wherein function (2) is performed using a combination of type A, type B, and customized modules.

C 1
6. (Previously amended) An expedited method of assembling a software system, comprising the following steps:

a) fabricating a collection of software systems, each of which contains

i) a processing module (PROC_MOD) which processes content of messages;

ii) a packaging module (PAK_MOD) which packages messages into packets for transport out of the system;

iii) a communication module (COM_MOD) which accepts and delivers message

09/550,192
Art Unit 3624
8446.00

packets; and

iv) a system control module (CONTROL) which coordinates the processes of (i), (ii), and (iii);

b) during the fabrication of paragraph (a),

i) fabricating identical CONTROL modules in all systems;

ii) fabricating identical COM_MOD modules in all systems;

iii) fabricating PAK_MOD modules in all systems, such that:

A) copies of a software unit A is contained in every PAK_MOD module;

B) some PAK_MOD modules contain a software unit B with no unit C; and

C) some PAK_MOD modules contain a software unit C with no unit B.

7. (Original) Method according to claim 6, and further comprising the following step:

iv) fabricating PROC_MOD modules in all systems, such that:

A) copies of a software unit D is contained in every PROC_MOD module;

09/550,192
Art Unit 3624
8446.00

- B) some PROC_MOD modules contain a software unit E with no unit F; and
- C) some PROC_MOD modules contain a software unit F with no unit E.

8. (New) Method according to claim 1, and further comprising the step of

- c) installing the software systems into electronic payment switches.

9. (New) Method according to claim 2, and further comprising the step of

- c) installing the software systems into electronic payment switches.

10. (New) Method according to claim 6, and further comprising the step of

- c) installing the software system into an electronic payment switch.

11. (New) Method according to claim 7, and further comprising the step of

- c) installing the software system into an electronic payment switch.

09/550,192
Art Unit 3624
8446.00

12. (New) Method according to claim 1, and further comprising:

- c) repeating steps of paragraph (b) to thereby modify a software system previously constructed; and
- d) installing the modified software system into an electronic payment switch.

13. (New) Method according to claim 6, and further comprising:

- c) repeating steps of paragraphs (a) and (b) to thereby modify a software system previously fabricated; and
- d) installing the modified software system into an electronic payment switch.